

In einem Proof-of-Concept kam Oracle Real-Time Decisions (RTD) zum Einsatz, um den Kunden eines Webshops zielgerichtetere Werbung präsentieren zu können. Dadurch wurde der Umsatz der begrenzten Werbeflächen vergrößert und gleichzeitig die User-Akzeptanz der Werbung erhöht.

Optimierung von Website-Werbung mit Oracle Real-Time Decisions

Thomas Kaiser, Riverland Reply GmbH

Als Nebenprodukt der Aktion fielen anonymisierte Informationen über das Verhalten bestimmter Nutzer- (Gruppen) ab, die für eine bessere Kundenansprache im klassischen Kampagnen-Marketing verwendet werden können. Hauptvorteile der Lösung sind jedoch der niedrige initiale Aufwand und die einfache Handhabung im Betrieb, da das manuelle Erarbeiten und Erstellen von Regeln in diesem Fall durch die selbstlernende RTD-Engine in weiten Teilen obsolet wurde.

Die Grundlagen

Oracle RTD ist eine Decision Engine, die in der Lage ist, nahezu in Echtzeit Entscheidungen auf der Basis sich im Zeitablauf optimierender Korrelationen zu treffen. Das unterscheidet sie

zum einen von regelbasierten Systemen, deren Logik vorab aufwändig definiert werden muss. Zum anderen grenzt sich RTD durch sein sofortiges Antwortverhalten von den meisten BI-Systemen ab, die zur Analyse meist Minuten bis Stunden benötigen. Durch die Verwendung offener Standards (wie XML, SOAP, Webservices, Java) erlaubt die Applikation gute Integrationsmöglichkeiten und einen Einsatz in vielfältigen Bereichen. Beispiele sind:

- Website Optimization
- Fraud Detection (am Geldautomaten oder im Internet)
- Call/Service Request Routing im Callcenter
- Customer Retention
- Agent-Support am Point-Of-Sale

- Cross Sell/Up Sell sowohl online als auch im Callcenter
- Online Personalization

Da RTD nur sehr schwer manuell trainiert werden kann, empfiehlt Oracle, die Evaluation und eine eventuell anschließende Einführung anhand eines sogenannten „Proof of Solution“ durchzuführen. Darunter ist eine voll funktionsfähige Lösung mit überschaubarem Funktionsumfang zu verstehen, die mit Produktivdaten beliefert wird. Auf dieser Basis können in folgenden Projekt-Phasen sowohl weitere Funktionalitäten in die bestehende Lösung integriert als auch komplett neue Applikationen (Cross-Channel) angebunden werden.

Das Projekt

Ein Unternehmen betreibt einen Webshop, der Werbeflächen für Produkte enthält, die gut zu den Kunden passen beziehungsweise ihnen gefallen könnten. Dieses Matching wird von Oracle RTD anhand verschiedener Attribute wie der Kaufhistorie, des Warenkorbs etc. durchgeführt. Zum besseren Verständnis wird zunächst die Architektur und anschließend ein Beispiel-Usecase beschrieben.

Die Architektur der RTD-Lösung stellt im ersten Schritt einen minimal-invasiven Ansatz dar, der nur die Einbindung einiger Komponenten (primär Skripte und Grafiken) in die bestehende Webseite und einen Export der verfügbaren Angebote des Webshops erfordert (siehe Abbildung 1). Dadurch ist eine schnelle und fast Release-unabhängige Umsetzung möglich. Die zentralen Komponenten sind

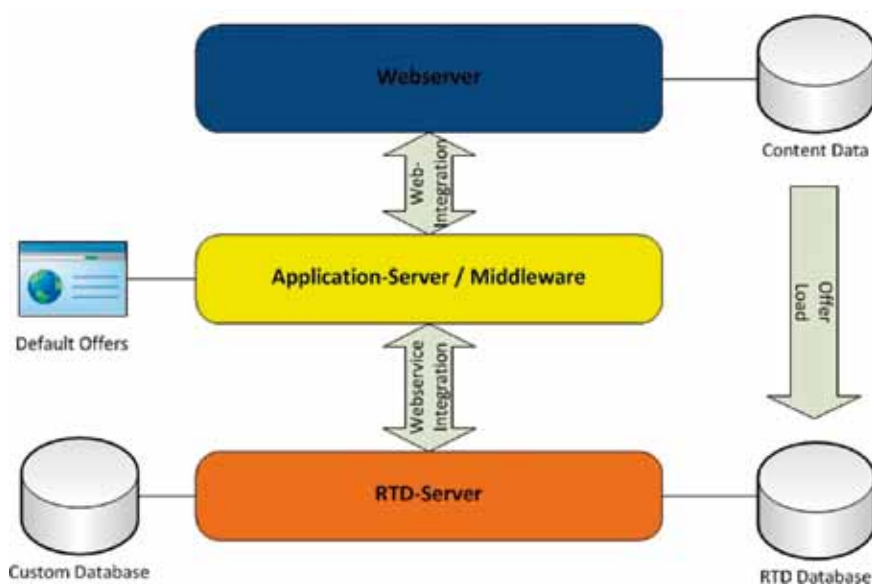


Abbildung 1: Die Systemarchitektur

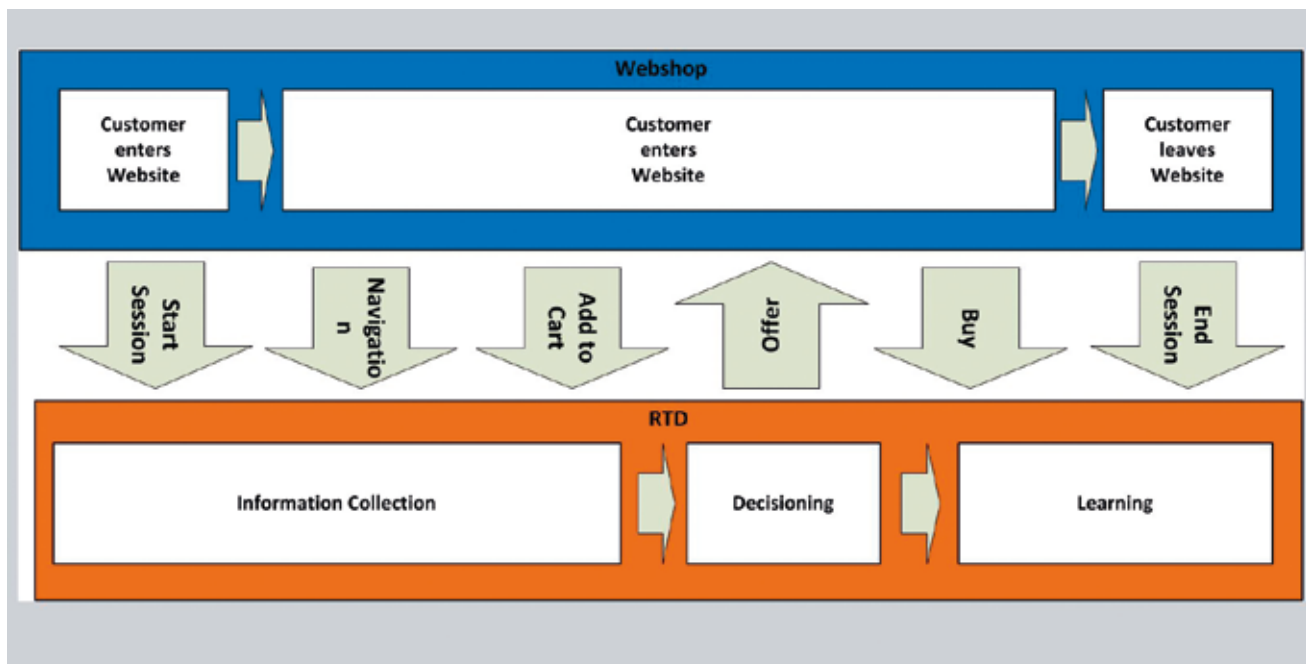


Abbildung 2: Beispielhafter Ablauf der Interaktionen

der Webserver (auf dem der bestehende Webshop läuft) und der RTD-Server (auf dem sämtliche RTD-Komponenten laufen), die beide über eine eigene Datenhaltung (Datenschutz) verfügen und durch einen zwischengeschalteten Application-Server verbunden sind. Abbildung 2 zeigt einen möglichen Ablauf der Interaktionen zwischen dem Webshop und der RTD-Applikation, die durch die Aktivitäten des Nutzers auf der Seite ausgelöst werden.

Mit dem Betreten der Seite beginnt die Aufzeichnung von Daten für diese Kunden-Session. Während der Kunde auf der Webseite navigiert und gegebenenfalls Produkte in den Warenkorb legt, werden bevorzugte Kategorien, Preisniveaus etc. registriert. Sobald eine Seite mit personalisierter Werbung aufgerufen wird, setzt der Webshop im Hintergrund einen Aufruf an RTD ab, das in Echtzeit die Angebote mit der größten Kaufwahrscheinlichkeit für diesen Kunden ermittelt und ihm präsentiert. Anschließend wird registriert, ob der Kunde das angebotene Produkt kauft, und auf Basis dieser Information der interne Lernprozess angestoßen. Das Verlassen der Seite schließt den Prozess und damit die Session ab. Die Reihenfolge der Schritte ist dabei natürlich vom Verhalten des Kunden abhän-

gig und somit in beliebiger Reihenfolge vorstellbar. Grundsätzlich muss RTD natürlich erst Informationen sammeln, um dem Kunden ein geeignete(re)s Angebot unterbreiten zu können.

Fazit

Die Vorteile der gewählten Architektur liegen auf der Hand: Mit verhältnismäßig geringem Initialaufwand lässt sich eine funktionierende Lösung relativ zeitnah (ein bis zwei Monate) anhand von Produktivdaten bezüglich ihres Mehrwerts evaluieren. Zur Bestimmung des Mehrwerts ist eine Kontrollgruppen-Funktionalität bereits in der Basis-Applikation integriert. Darauf aufbauend können sowohl der Funktionsumfang der bestehenden Lösung erweitert als auch neue Kanäle (wie Callcenter, Point of Sale etc.) angebunden werden. Darüber hinaus ist die Verwendung der Daten für eine Website-Personalisierung zur Laufzeit vorstellbar, die Kunden-individuelle Navigations-Menüs erlaubt. Zudem lassen sich die statistischen Korrelationen, die für den Entscheidungsprozess verwendet und ständig aktualisiert werden, zum einen für Analysen des Kaufverhaltens oder in einem weitergehenden Schritt beispielsweise auch für E-Mail-Kampagnen einsetzen. Für den Kun-

den ergibt sich der Vorteil, speziell auf ihn zugeschnittene Angebote zu erhalten, anstatt mit einer Flut unpassender Werbung überschüttet zu werden. Dies kann in der Folge zu mehr Umsatz und auch zu Loyalität führen, da der Kunde sich eher verstanden fühlt.

Das (speziell in Deutschland) oft angesprochene Thema „Datenschutz“ muss natürlich in allen Phasen des Projekts beachtet werden, was aber aufgrund der Applikations-Architektur kein Problem darstellen sollte, denn zum einen speichert die RTD-Datenbank keine Inter-beziehungsweise Trans-Aktionsdatensätze, sondern arbeitet (schon aus Performance-Gründen) nur mit aggregierten Summendaten. Zum anderen ist die eindeutige Identifikation eines Kunden gar nicht notwendig, da der verwendete Fuzzy-Ansatz primär mit ähnlichen Mustern (hier Käufergruppen) und nicht mit eindeutigen Zuordnungen (wie Kundennummer) arbeitet.



Thomas Kaiser
t.kaiser@reply.de